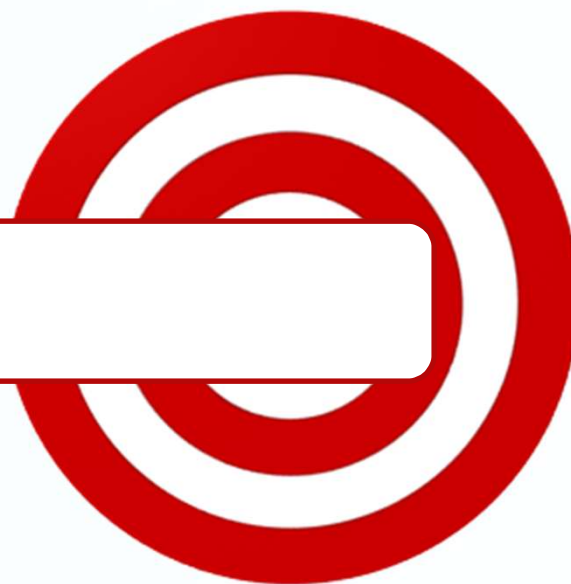


Apache NiFi



APACHE NIFI



Apache NiFi

NiFi was built to automate the flow of data between systems.

- While the term 'dataflow' is used in a variety of contexts, we use it here to mean the automated and managed flow of information between systems.
- This problem space has been around ever since enterprises had more than one system, where some of the systems created data and some of the systems consumed data.



Challenges of Data Flow

- Systems fail
- Data access exceeds capacity to consume
- Boundary conditions are mere suggestions
- What is noise one day becomes signal the next
- Systems evolve at different rates
- Compliance and security
- Continuous improvement occurs in production



Key Features of NiFi

- Guaranteed delivery
 - Data buffering
 - Backpressure
- Pressure release
- Prioritized queuing
- Flow specific QoS
 - Latency vs. throughput
 - Loss tolerance
- Data provenance
- Supports push and pull models
- Recovery/recording a rolling log of fine-grained history
- Visual command and control
- Flow templates
- Pluggable, multi-tenant security
- Designed for extension



NiFi Terminology

Core Components of NiFi

Flow File

FlowFile Processor

Connection

Flow Controller

Process Group



FlowFile

- Each piece of "User Data" (i.e., data that the user brings into NiFi for processing and distribution) is referred to as a FlowFile.
- A FlowFile is made up of two parts:
 - Attributes
 - Content.
- The Content is the User Data itself.
- Attributes are key-value pairs that are associated with the User Data.



FlowFile

FlowFile

Standard FlowFile Attributes

Key: 'entryDate' Value: 'Fri Jun 17 17:15:04 EDT 2016'

Key: 'lineageStartDate' Value: 'Fri Jun 17 17:15:04 EDT 2016'

Key: 'fileSize' Value: '23609'

FlowFile Attribute Map Content

Key: 'filename' Value: '15650246997242'

Key: 'path' Value: './'

Header

*Binary Content **

Content

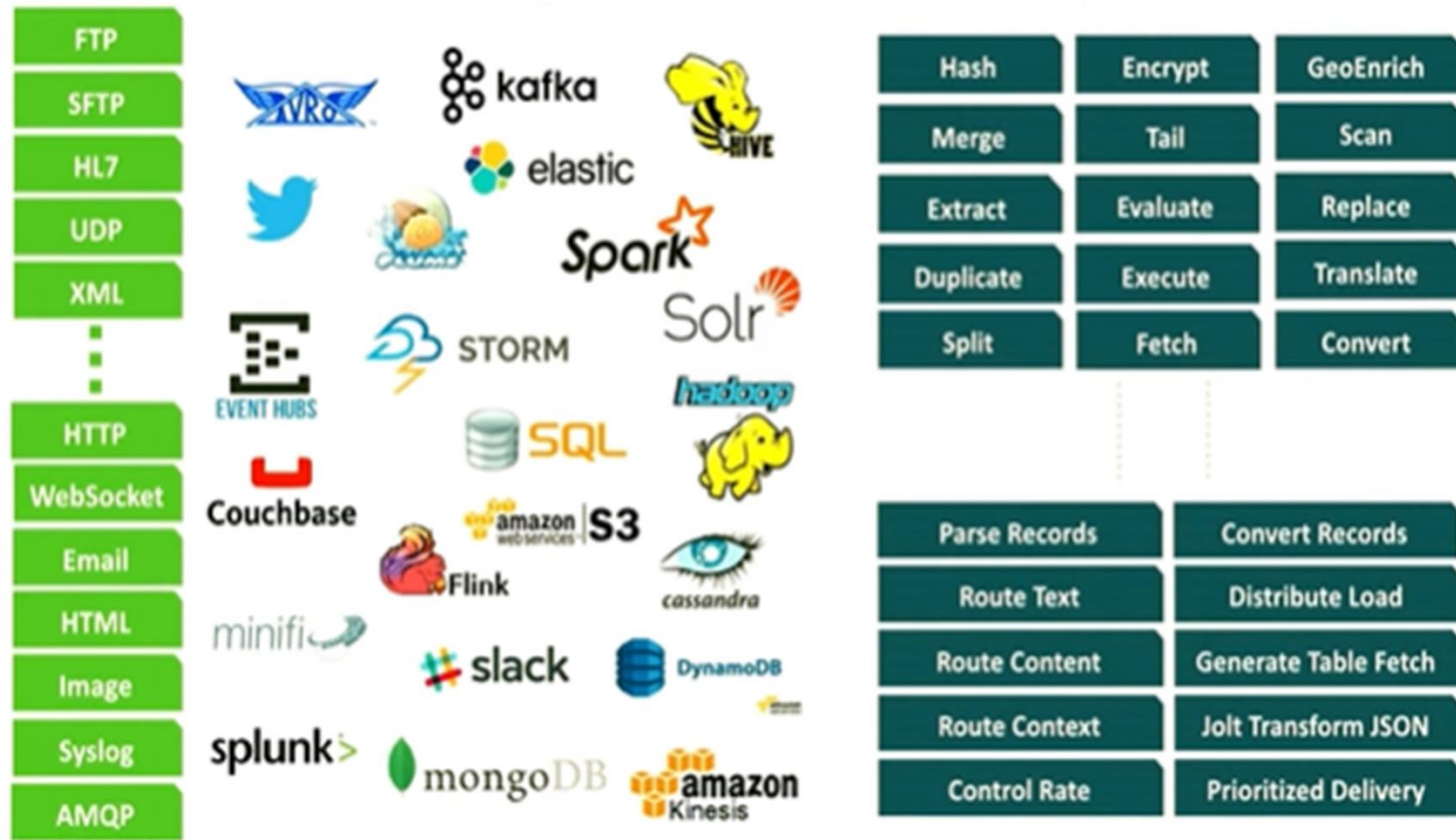


FlowFile Processors

- Processors actually perform the work doing some combination of data routing, transformation, or mediation between systems.
- Processors have access to attributes of a given FlowFile and its content stream.
- Processors can operate on zero or more FlowFiles in a given unit of work and either commit that work or rollback.
- NiFi provides over 260 out-of-the-box processors and 48 controller services.
- In addition, developers can write their own custom processors.



FlowFile Processors



Processors Categories

There are 286 (in version 1.11) bundled processors and many more open source processors.

Data Transformation: ReplaceText, JoltTransformJSON...

Routing and Mediation: RouteOnAttribute, RouteOnContent, ControlRate...

Database Access: ExecuteSQL, ConvertJSONToSQL, PutSQL...

Attribute Extraction: EvaluateJsonPath, ExtractText, UpdateAttribute...

System Interaction: ExecuteProcess ...

Data Ingestion: GetFile, GetFTP, GetHTTP, GetHDFS, ListenUDP, GetKafka...

Sending Data: PutFile, PutFTP, PutKafka, PutEmail...

Splitting and Aggregation: SplitText, SplitJson, SplitXml, MergeContent...

HTTP: GetHTTP, ListenHTTP, PostHTTP...

AWS: FetchS3Object, PutS3Object, PutSNS, GetSQS



Connections

- Connections provide the actual linkage between processors.
- These act as queues and allow various processes to interact at differing rates.



Flow Controller

- The Flow Controller maintains the knowledge of how processes connect and manages the threads and allocations thereof which all processes use.
- The Flow Controller acts as the broker facilitating the exchange of FlowFiles between processors.



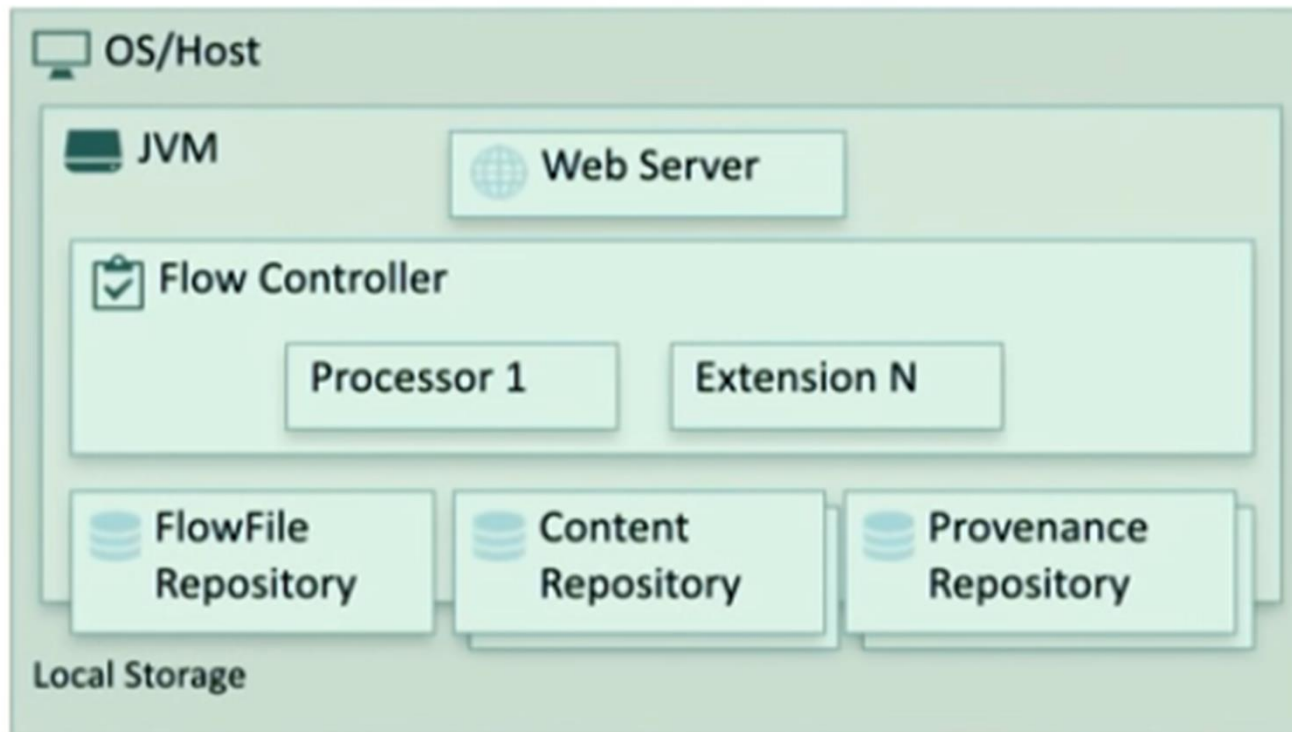
Process Group

- A Process Group is a specific set of processes and their connections, which can receive data via input ports and send data out via output ports.
- In this manner, process groups allow creation of entirely new components simply by composition of other components.

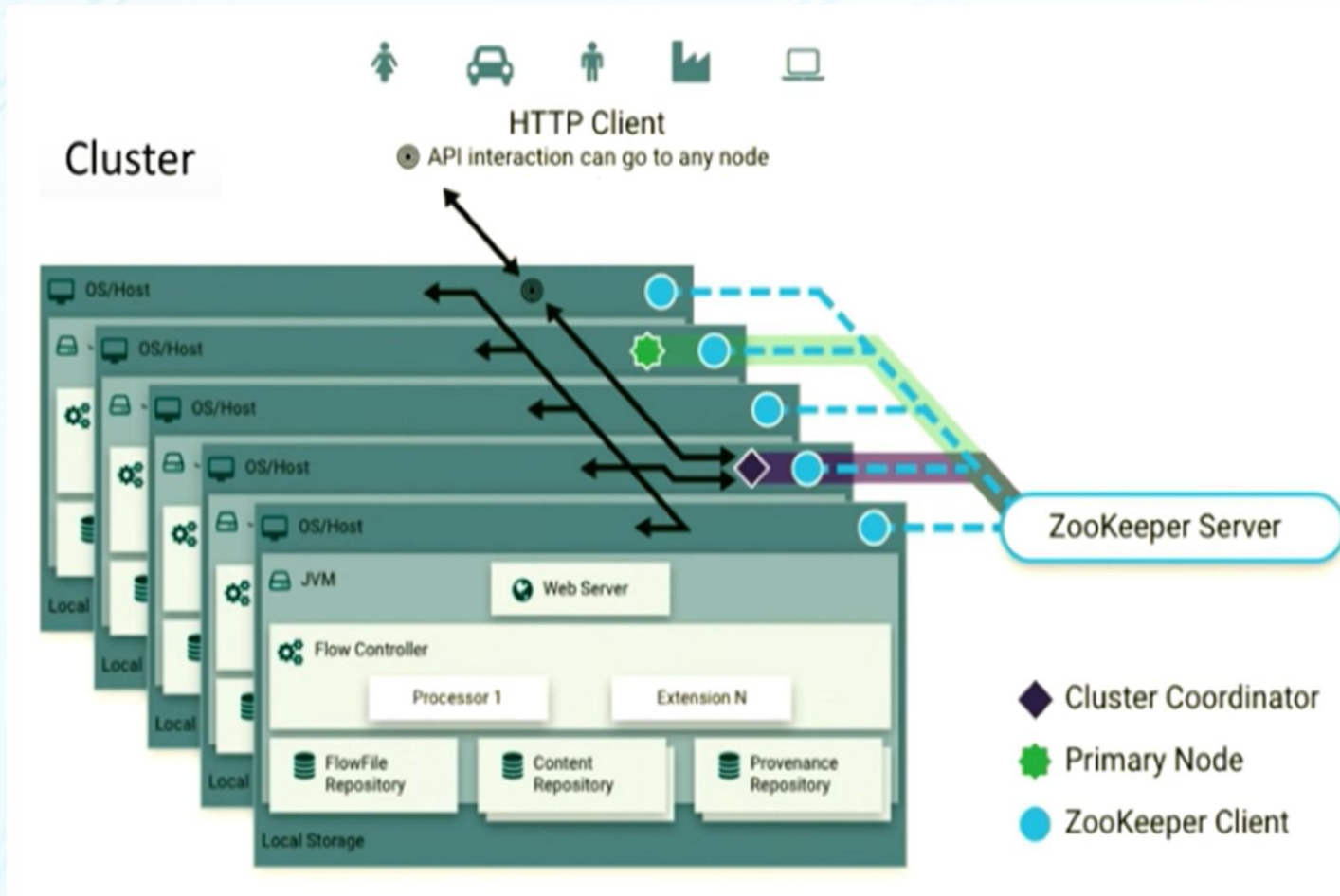


NiFi Architecture

Standalone



NiFi Architecture

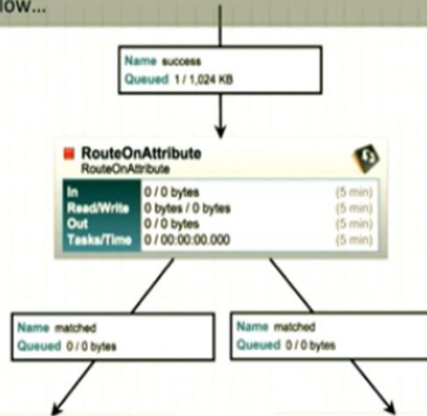


Repositories - Pass by Reference

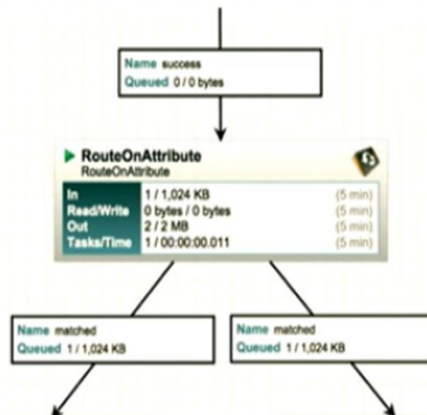
Excerpt of demo flow...

What's happening inside the repositories...

BEFORE



AFTER



$F_1 \rightarrow C_1$

C_1

$P_1 \rightarrow F_1$

$F_1 \rightarrow C_1$


$F_2 \rightarrow C_1$

C_1


$P_1 \rightarrow F_1 - \text{Create}$

$P_2 \rightarrow F_1 - \text{Route}$

$P_3 \rightarrow F_2 - \text{Clone } (F_1)$

 FlowFile

 Content

 Provenance

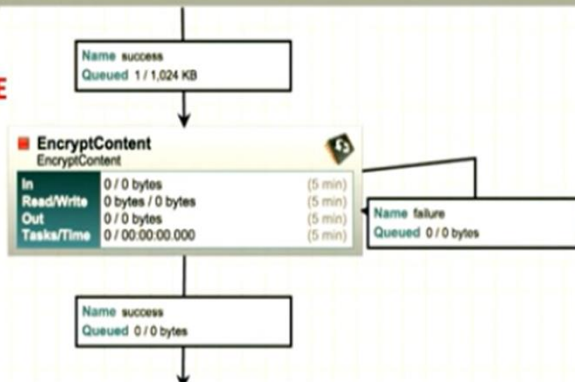


Repositories - Copy on Write

Excerpt of demo flow...

What's happening inside the repositories...

BEFORE

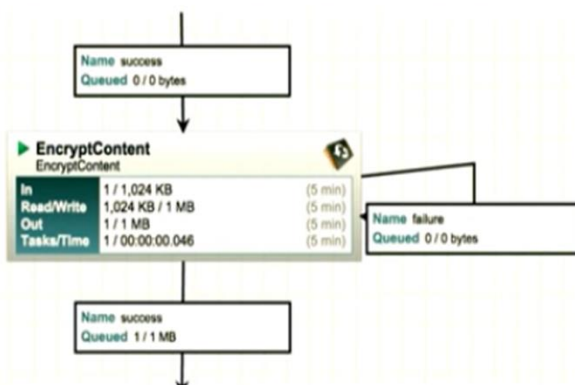


$F_1 \rightarrow C_1$

C_1

$P_1 \rightarrow F_1 - \text{CREATE}$

AFTER



$F_1 \rightarrow C_1$
 $F_{1.1} \rightarrow C_2$

C_1 (plaintext)
 C_2 (encrypted)

$P_1 \rightarrow F_1 - \text{CREATE}$
 $P_2 \rightarrow F_{1.1} - \text{MODIFY}$

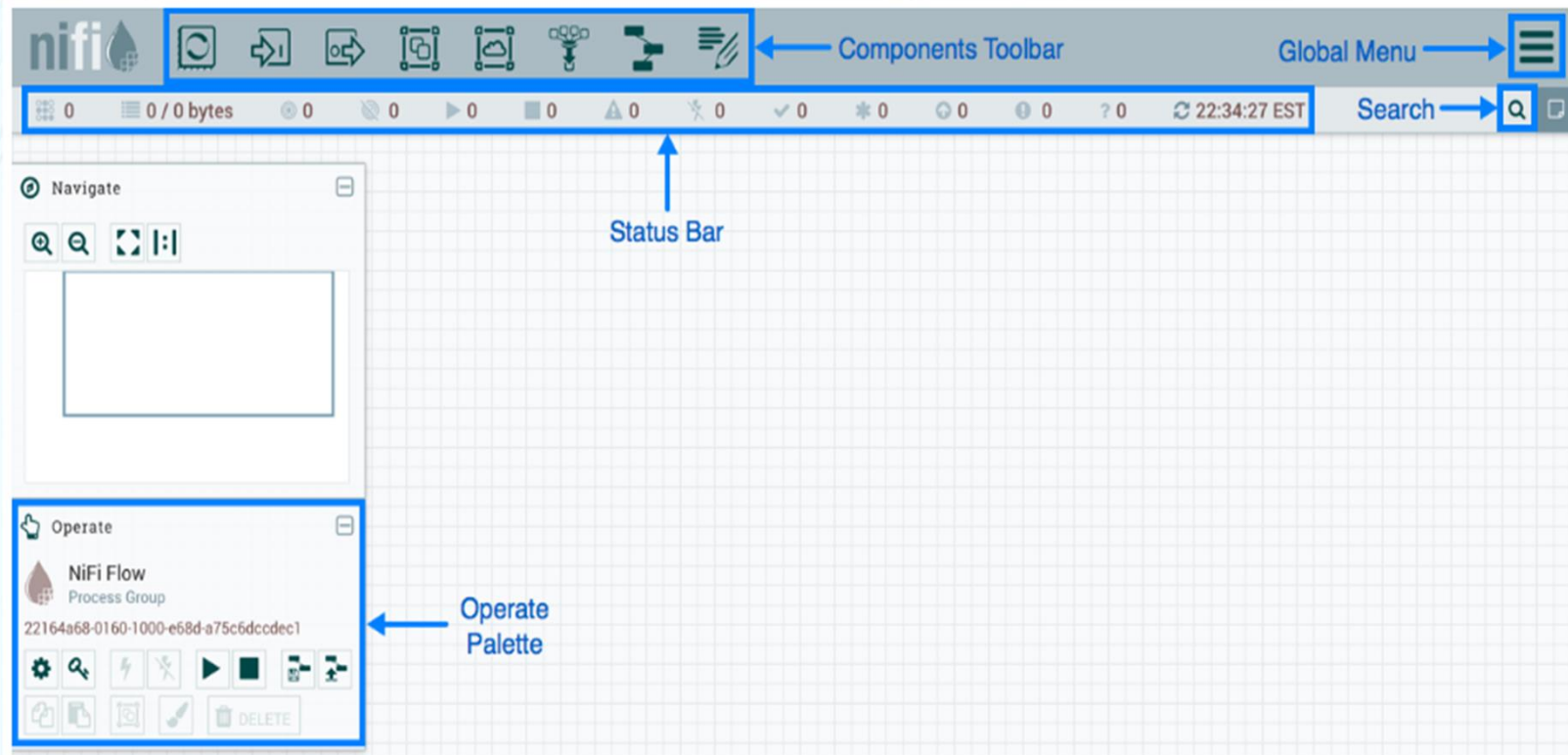
FlowFile

Content

Provenance



NiFi UI



Add a Processor

- We can begin creating our dataflow by adding a Processor to our canvas.
- To do this, drag the Processor icon from the top-left of the screen into the middle of the canvas (the graph paper-like background) and drop it there.
- This will give us a dialog that allows us to choose which Processor we want to add



Add a Processor

Add Processor

Source

all groups ▼

amazon attributes
avro aws consume
csv database fetch
files get hadoop
ingest input insert
json listen logs
message put
remote restricted
source sql text
update

Displaying 219 of 219

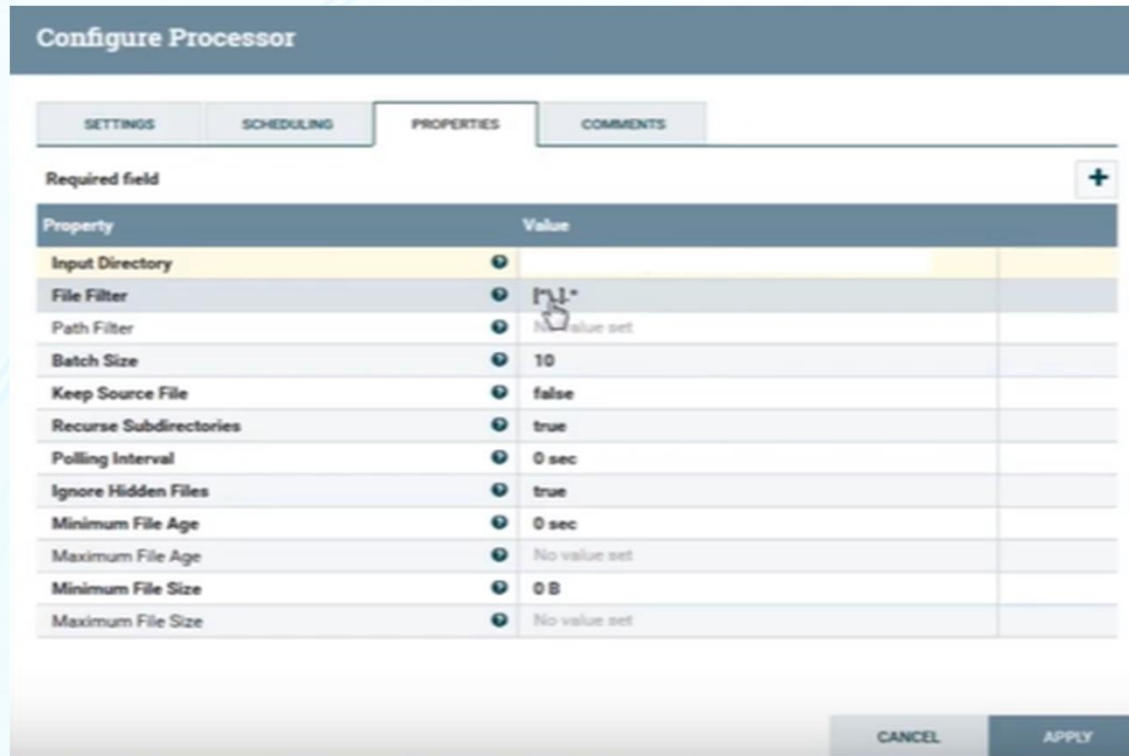
Filter

Type ▲	Version	Tags
AttributeRollingWindow	1.2.0	rolling, data science, Attribute Expression Language, st...
AttributesToJSON	1.2.0	flowfile, json, attributes
Base64EncodeContent	1.2.0	encode, base64
CaptureChangeMySQL	1.2.0	cdc, jdbc, mysql, sql
CompareFuzzyHash	1.2.0	fuzzy-hashing, hashing, cyber-security
CompressContent	1.2.0	lzma, decompress, compress, snappy framed, gzip, sna...
ConnectWebSocket	1.2.0	subscribe, consume, listen, WebSocket
ConsumeAMQP	1.2.0	receive, amqp, rabbit, get, consume, message
ConsumeEWS	1.2.0	EWS, Exchange, Email, Consume, Ingest, Message, Get,...
ConsumeIMAP	1.2.0	Imap, Email, Consume, Ingest, Message, Get, Ingress
ConsumeJMS	1.2.0	jms, receive, get, consume, message
ConsumeKafka	1.2.0	PubSub, Consume, Inqest, Get, Kafka, Ingress, Topic, 0...



Configure a Processor

- After adding the Processor, we can configure it by right-clicking on the Processor and choosing the Configure menu item.
- Here we add the attributes related to that processor.



The screenshot shows the 'Configure Processor' dialog box with the 'PROPERTIES' tab selected. The dialog has four tabs: SETTINGS, SCHEDULING, PROPERTIES, and COMMENTS. Below the tabs is a 'Required field' label with a '+' icon. The main area contains a table with two columns: 'Property' and 'Value'. The 'File Filter' property is highlighted, and its value is '.*'. A mouse cursor is pointing at the 'File Filter' row. The 'Batch Size' is set to '10', 'Keep Source File' is 'false', 'Recurse Subdirectories' is 'true', 'Polling Interval' is '0 sec', 'Ignore Hidden Files' is 'true', 'Minimum File Age' is '0 sec', 'Maximum File Age' is 'No value set', 'Minimum File Size' is '0 B', and 'Maximum File Size' is 'No value set'. At the bottom right are 'CANCEL' and 'APPLY' buttons.

Property	Value
Input Directory	
File Filter	.*
Path Filter	No value set
Batch Size	10
Keep Source File	false
Recurse Subdirectories	true
Polling Interval	0 sec
Ignore Hidden Files	true
Minimum File Age	0 sec
Maximum File Age	No value set
Minimum File Size	0 B
Maximum File Size	No value set

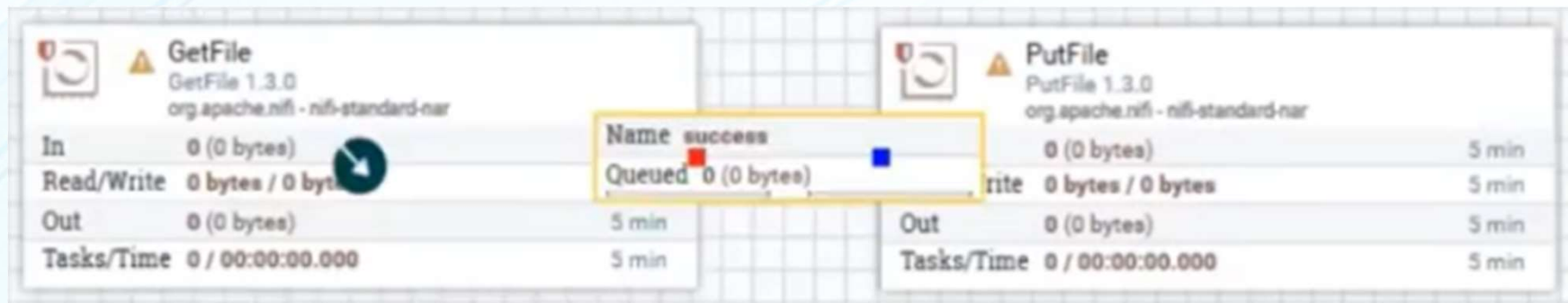


Connect Processors

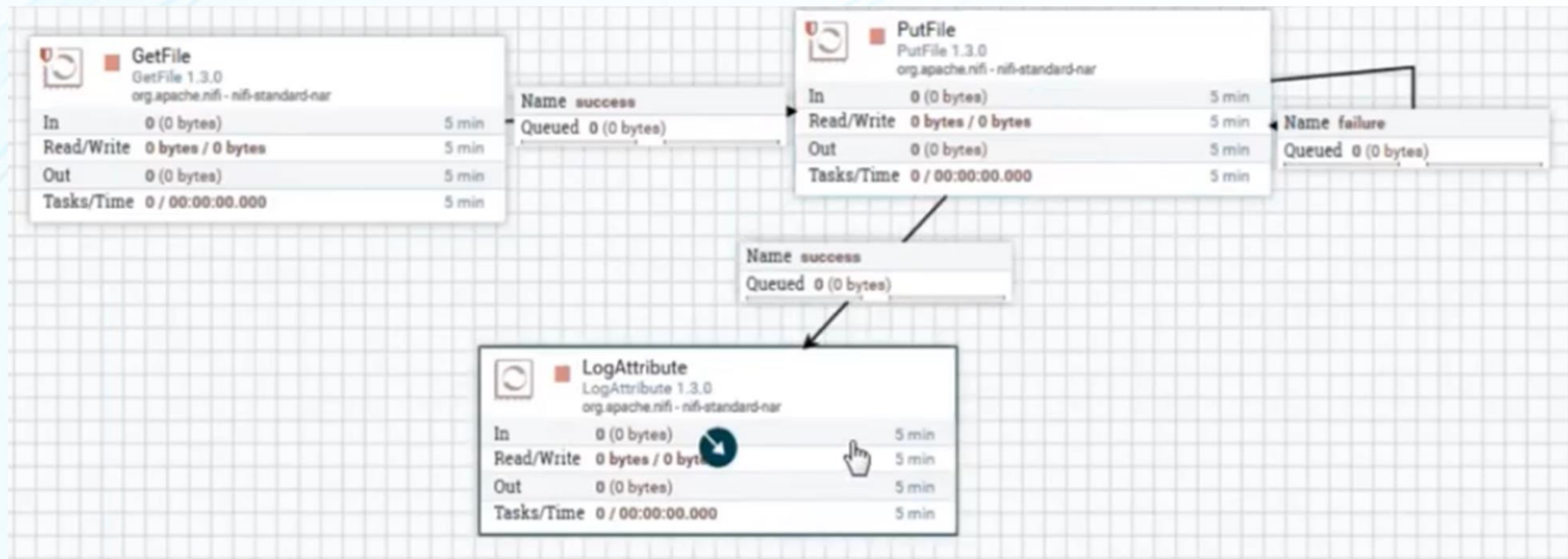
- Each Processor has a set of defined "Relationships" that it is able to send data to.
- When a Processor finishes handling a FlowFile, it transfers it to one of these Relationships. This allows a user to configure how to handle FlowFiles based on the result of Processing.
- For example, many Processors define two Relationships: success and failure. Users are then able to configure data to be routed through the flow one way if the Processor is able to successfully process the data and route the data through the flow in a completely different manner if the Processor cannot process the data for some reason. Or, depending on the use case, it may simply route both relationships to the same route through the flow.



Connect Processors



Full View of a Workflow



THANK YOU

